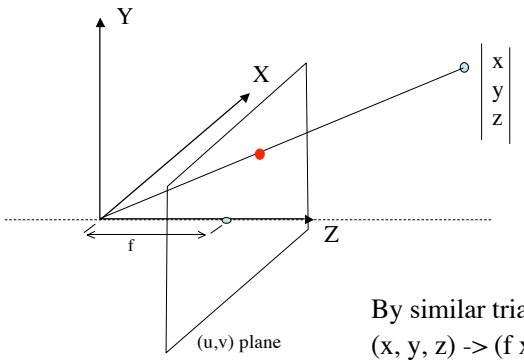
Perspective example (onto z=f)



By similar triangles,
(x, y, z) -> (f x/z, f y/z, f)

The equation of projection

- In homogeneous coordinates

$$(x,\ y,\ z,\ 1) \rightarrow (f\frac{x}{z},\ f\frac{y}{z},\ 1)$$

- Equivalently

$$(x,\ y,\ z,\ 1) \rightarrow (x,\ y,\ \frac{z}{f})$$

- Homogeneous coordinates are being used to store foreshortening

- Note that using regular coordinates does **not** yield a linear transformation (inconvenient!).

The projection matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{bmatrix}$$

Assembling the pieces

$$\begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{matrix} \text{Transformation} \\ \text{representing} \\ \text{intrinsic parameters} \end{matrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{matrix} \text{Transformation} \\ \text{representing} \\ \text{extrinsic parameters} \end{matrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Camera matrix, M

Note that we assume that the intrinsic parameters absorb the focal length (so it is one in the projection matrix)

First part makes it so that we are in standard camera coords where we know how to project.

Cameras so far

- Model for the brightness due to light reaching a region (say, CCD element)

$$(R,G,B) = \int_{380}^{780} \; * \; d\lambda$$

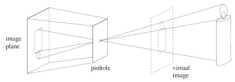
$$\square^{(k)} = F^{(k)}(\square^{(k)}) = F^{(k)}\left(\square L(\square)R^{(k)}(\square)d\square\right)$$

or

$$\square^{(k)} = \mathbf{L} \bullet \mathbf{R}^{(k)}$$

- Spectral camera calibration task is to determine R (and F) from data.

- Model the image location corresponding to a point in the world by projection (developed using pinhole camera model).



- Represent using matrix multiplication using homogenous coordinates
- (u,v,w) = **M** * (X, Y, Z, 1)
- Geometric camera calibration task is to determine **M** from data.

Supplementary material

Real cameras

(Supplementary material on these topics posted on-line)

- Real cameras need lenses
  - Focus now depends on distance (unlike pinhole cameras)
  - Various aberrations and distortions
    - E.g. Chromatic aberration
- Brightness falls off towards edges
  - Fall off due to projection onto flat surface
  - Vignetting
- Scattering at optical surfaces (flare)
- Capture process has many sources of noise

Linear Least Squares (§3.1)

- Very common problem in vision: solve an over-constrained system of linear equations
  - e.g., Ux=y, where U has more rows than needed
- More equations allows multiple measurements to be used
- Least squares means that you minimize squared error (the difference between your model and your data)
- Least squares minimization is (relatively) easy
- Not very robust to outliers (assumes error is Gaussian)
- Used, overused, and abused.

Linear Least Squares (§3.1)

We will look at two problems

First, $U\mathbf{x} = \mathbf{y}$ where U has more rows than needed

Second, $U\mathbf{x} = \mathbf{0}$ subject to $|\mathbf{x}| = 1$ were U has more rows than needed

We will use the **second** problem for geometric camera calibration.

We will start with the **first** (a bit easier)

Math aside, #3

Linear Least Squares (§3.1)

Problem one $U\mathbf{x} = \mathbf{y}$ where U has more rows than needed

U is not square, so inverting it does not work

(Following solution sketch is optional, but try to get main idea)

Define $\mathbf{e} = U\mathbf{x} - \mathbf{y}$ and $E = |\mathbf{e}|^2 = \mathbf{e}^T\mathbf{e}$

Least squares solution is defined by finding the **x** that gives the minimum value for E (by differentiating by each x_i, and setting all resulting equations to zero).

Details optional

Linear Least Squares (§3.1)

$$\frac{\partial E}{\partial x_i} = 2\frac{\partial \mathbf{e}^T}{\partial x_i}\mathbf{e} = 0 \quad \text{(for minimum)}$$

This is true for all components, x_i

Details optional

Linear Least Squares (§3.1)

$$\frac{\partial E}{\partial x_i} = 2\frac{\partial \mathbf{e}^T}{\partial x_i}\mathbf{e} = 0 \quad \text{(for minimum)}$$

This is true for all components, x_i, so we get:

$$\begin{bmatrix} \cdots & \\ & \\ \dfrac{\partial \mathbf{e}^T}{\partial x_i} & \\ & \\ \cdots & \end{bmatrix} \mathbf{e} = 0$$

Details optional

Linear Least Squares (§3.1)

Evaluating $\dfrac{\partial \mathbf{e}^T}{\partial x_i}$

By definition $\dfrac{\partial \mathbf{e}^T}{\partial x_i} = \left[\dfrac{\partial}{\partial x_i}(U\mathbf{x} - \mathbf{y})\right]^T = \left[\dfrac{\partial}{\partial x_i}U\mathbf{x}\right]^T$

$U\mathbf{x}$ is a column vector with elements that look like

$$(U\mathbf{x})_k = \sum_j U_{kj}x_j$$

Details optional

Linear Least Squares (§3.1)

$(U\mathbf{x})_k = \sum_j U_{kj} x_j$

So $\frac{\partial}{\partial x_i}(U\mathbf{x})_k = U_{ki}$

And so $\frac{\partial}{\partial x_i} U\mathbf{x}$ is the i'th column of U

And so $\frac{\partial \mathbf{e}^T}{\partial x_i} = \left[\frac{\partial}{\partial x_i} U\mathbf{x}\right]^T$ is the i'th row of U^T

Details optional

Linear Least Squares (§3.1)

$\frac{\partial \mathbf{e}^T}{\partial x_i} = \left[\frac{\partial}{\partial x_i} U\mathbf{x}\right]^T$ is the i'th row of U^T

So $\begin{bmatrix} \dots \\ \frac{\partial \mathbf{e}^T}{\partial x_i} \\ \dots \end{bmatrix} \mathbf{e} = 0$ becomes $U^T(U\mathbf{x} - \mathbf{y}) = 0$

Details optional

Linear Least Squares (§3.1)

From the previous slide our condition is $U^T(U\mathbf{x} - \mathbf{y}) = 0$

Or $U^T U\mathbf{x} = U^T\mathbf{y}$

So $\mathbf{x} = (U^T U)^{-1} U^T\mathbf{y}$ Note that $(U^T U)^{-1}$ can easily be shown to exist

Thus

$\mathbf{x} = U^\dagger\mathbf{y}$ where $U^\dagger = (U^T U)^{-1} U^T$ is the pseudoinverse of U

Important

Non-homogeneous linear least squares summary
(the part you need to know)

You should be able to set up

$U\mathbf{x} = \mathbf{y}$

You should know that it is solved by

$\mathbf{x} = U^\dagger\mathbf{y}$ where $U^\dagger$ is the pseudoinverse of U

You can assume that you can look up

$U^\dagger = (U^T U)^{-1} U^T$

*You should also keep in mind that for numerical stability, one may have to use a different approach to solve (without matrix inversion) the following

$U^T U\mathbf{x} = U^T\mathbf{y}$

Important

Non-homogeneous linear least squares
(example one---naïve spectral camera calibration)

Remember the fact that the camera has a spectral sensitivity R(□). So how do we find it out?

Recall that $\square = \int L(\square)R(\square)d\square$

has the discrete version

$$\square = \mathbf{L} \bullet \mathbf{R}$$

(previously we accounted for multiple channels with the superscript (k), but here we just consider each channel separately)

Important

Non-homogeneous linear least squares
(example one---naïve spectral camera calibration)

Strategy: measure some spectra entering the camera, $\mathbf{L}_i$, and note the response, $\square_i$.

So we have, for a bunch of measurements, i:

$\square_i = \mathbf{L}_i \bullet \mathbf{R}$

If we don't have enough measurements, then the problem is under constrained. To account for noise, we want to use multiple measurements.

Important

Non-homogeneous linear least squares
(example one---naïve spectral camera calibration)

From:

$\square_i = \mathbf{L}_i \bullet \mathbf{R}$

The path is clear. Just form a matrix L with rows $\mathbf{L}_i$, a vector **P** with elements $\square_i$, and solve the least squares equation

$L\mathbf{R} = \mathbf{P}$